

WATERTIGHT ANISOTROPIC SURFACE MESHING USING QUADRILATERAL PATCHES

Robert Haimes¹

Michael J. Aftosmis²

¹*Massachusetts Institute of Technology, Cambridge, MA, U.S.A. haimes@mit.edu*

²*NASA Ames Research Center, Moffett Field, CA, U.S.A. aftosmis@nas.nasa.gov*

ABSTRACT

This paper presents a simple technique for generating anisotropic surface triangulations using a generic unstructured quadrilateral decomposition of CAD entities that map to a logical rectangle. Watertightness and geometric quality measures are maintained and are consistent with those obtained using CAPRI's default tessellation algorithm. Output surface triangulations using the new method meet user specified criteria for chord-height, neighbor triangle dihedral angle, and maximum triangle side length. This discrete representation has *hooks* back to the owning geometry and therefore can be used in conjunction with these entities to allow for easy enhancement or modification of the tessellation suitable for grid generation or other downstream applications.

Keywords: surface triangulation, anisotropy, quadrilateral, CAD geometry

1. INTRODUCTION

The premier design goal for the Computational Analysis **PR**ogramming Interface (CAPRI) [1] is that geometry access be appropriate for CAE developers. Early in the design and implementation of CAPRI, it became obvious that providing an Application Programming Interface (API) that only gives the programmer access to the geometry and topology of a solid part was insufficient. The burden of deciphering the CAD data and attempting to generate a discrete representation of the surfaces required for mesh generation was too great. Fortunately, many grid generation systems (used in CFD and other disciplines) can use watertight surface triangulations as input. Combining a discretized view of the solid part as well as its geometry and topology can provide a complete, and easier to use, access point into the CAD data. A tessellation of the object that contains not only the mesh coordinates and supporting triangle indices but other data, such as the underlying CAD surface parameters (for each point), as well as the connectivity of the triangles, assists in traversing through and dissecting the CAD representation of a part.

An important aspect of CAPRI is that it provides CAD vendor neutral access to all of the data obtained from the models that can be passed back to the application. The triangulation generated by CAPRI is guaranteed to be watertight, regardless of the CAD kernel in use. Some CAD system geometry kernels can provide data of this quality (i.e., UniGraphics, Parasolid, CATIA and ComputerVision). Other CAD systems can provide the data, but it is not of sufficiently high quality to use. (For example, Pro/Engineer requires one to buy Pro/MESH to get a closed triangulation.) Finally, SDRC's Open I-DEAS API does not provide access to a triangulation at all. The fact that not all CAD systems provide such a tessellation has forced the development of a surface triangulator within CAPRI for CAD solid parts that *does* meet all of the quality requirements [3].

It should be noted that CAPRI's tessellations are not intended as the starting point for computational analysis (though they could be used in some cases). Since CAPRI sees only geometry it cannot anticipate the smoothness, resolution or other requirements of downstream application. Output triangulations approxi-

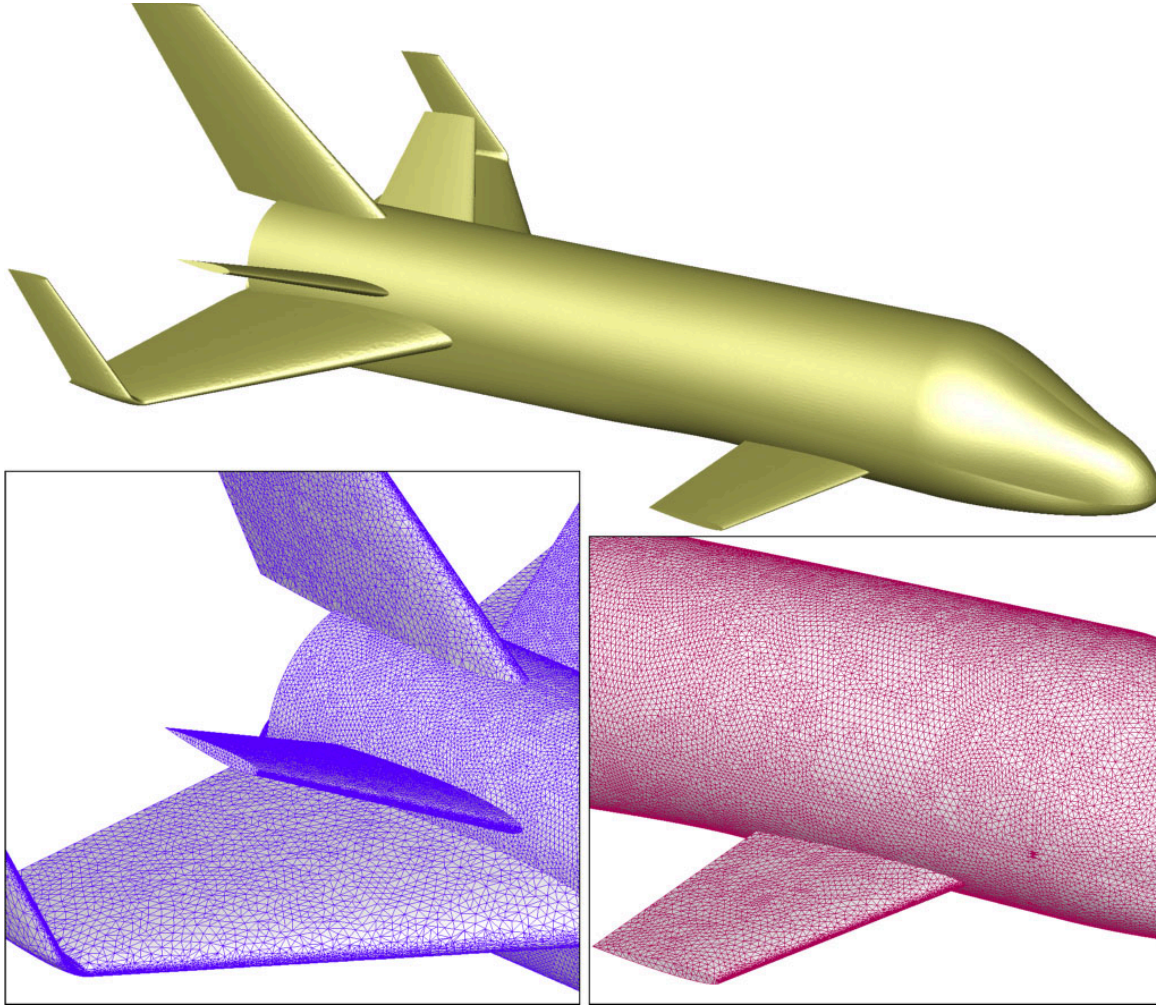


Figure 1: CAPRI's Isotropic Triangulation for the Reusable Launch Vehicle's notional geometry.

mate the geometry only and some processing of the tessellation is expected in order to match this triangulation with the requirements of the physical problem being investigated. The triangulation can be enhanced through either physical or parameter space manipulation, using point “snap” and (u, v) surface evaluations routines provided by the CAPRI API [2].

In 2002 [3] we presented an overview of CAPRI's implementation of the quality triangulation technique first developed in [2] which produces watertight triangulations for any input CAD solid. An example of the use of this scheme can be seen in Figure 1. This tessellation of a notional Reusable Launch Vehicle (RLV) is comprised of an assembly of 11 solids containing a total of 161 CAD Faces. This triangulation matches pre-specified criteria for chord-height deviation, maximum allowable dihedral angle, and maximum edge length. The example in Figure 1 took the CAD-native part and assembly files as input. The output triangulation shown has about 700k triangles, and was generated

completely *hands-off*.

CAPRI's implementation guarantees several characteristics of the output triangulation:

- **Robust.** It is imperative that the scheme works for all possible topologies and provides a tessellation that can be used.
- **Correct.** The triangulation is of no use if it is not true to the CAD model. The tessellation must be logically correct; i.e. provide a valid triangulation in the parameter space (u, v) of the individual surface. It must also be geometrically correct; i.e. depict a surface triangulation that truly approximates the geometry. This involves ensuring all facets have a consistent orientation with no creases or abrupt changes in triangle normals. Correctness in both physical and parameter space allows CAPRI based application enhancement schemes to operate in either or both.

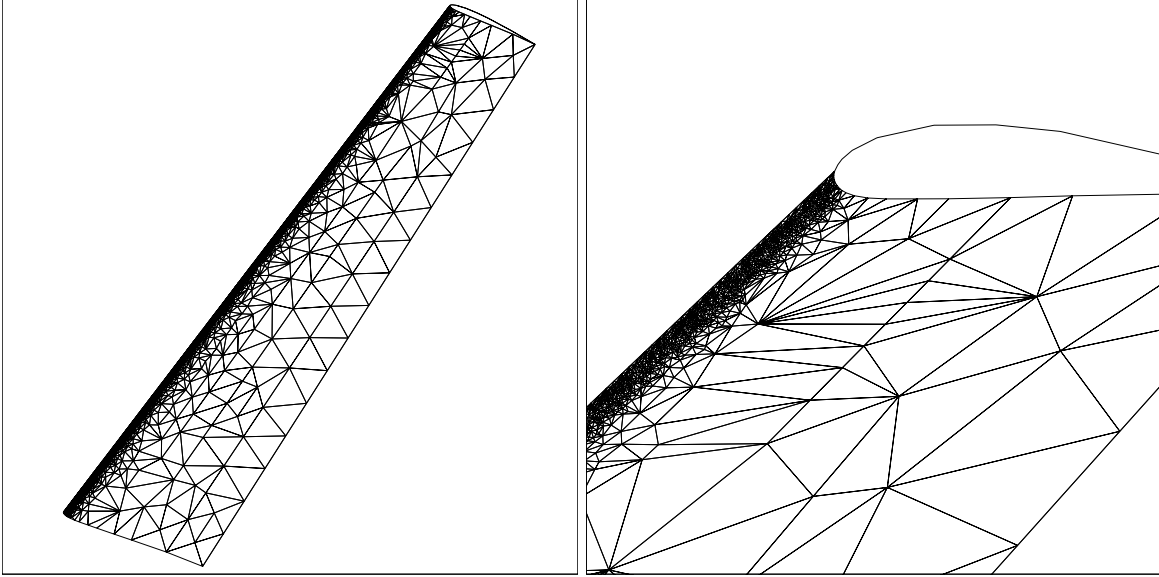


Figure 2: Isotropic Triangulation using default Tessellation scheme. #points = 6424, #triangles = 12639, CPU time = 3.8 seconds

- Adjustable. To minimize the post-processing of CAPRI's tessellation for a specific discipline or analysis, some *a priori* adjustment of the resultant quality is available. It must be noted however, that any criteria may not be met (especially near the bounds of a CAD object) due to issues of closure and solid model accuracy. This goal may conflict with the more important characteristic of being watertight and having a smooth surface representation. The parameters are:
 - Maximum triangle side length. Any triangle sides (not on a CAD Edge) longer (in (x, y, z)) than a specified value are bisected.
 - Maximum dihedral-angle between two triangles. Any two triangles on the same CAD Face whose (x, y, z) facet normals differ by more than the input value will be broken up.
 - Chord-height tolerance. When the deviation between triangle center and actual surface (in (x, y, z)) is greater than the specified value then the triangle is subdivided by inserting the center point.
- No geometric translation. To truly facilitate hands-off grid generation, anything that requires user intervention must be avoided. All data maintained within CAPRI is consistent with the CAD's solid model representation. An alternate or translated representation is not used, because then the result will be something different than resides within CAD.
- Watertight. Triangulated CAD solids are closed and conformal; having this characteristic allows for meshing without “fixing” geometry. For the tessellation of a solid object, this means that all Edge (trimming) curves terminate at consistent coordinates of the bounding Nodes and a single discretization for Edge curves be used on both surfaces sharing the common Edge. Each triangle side in the tessellation is shared by exactly two triangles, and the star of each vertex is surrounded and bounded by a single closed loop of sides. The triangulation is everywhere locally manifold. In a manifold triangulation, there are no voids, cracks or overlaps of any triangles that make up the solid.
- Smoothness. It should be noted that inserted points are neither moved nor removed. Once the quality metrics described above are met, the algorithm stops. There is no attempt made to have the triangulations meet any measure in the *eye-ball norm*.

As a counterpoint to Figure 1, Figure 2 shows an isolated aircraft wing tessellated using this system. This example illustrates both the strengths and weaknesses of the approach. The input geometry in this case was a single CAD solid, in its native format, and the tessellation shown was produced using the default input parameters. The tessellation is watertight, and adjacent triangles satisfy the chord-height, dihedral-angle and maximum edge length requirements. Since

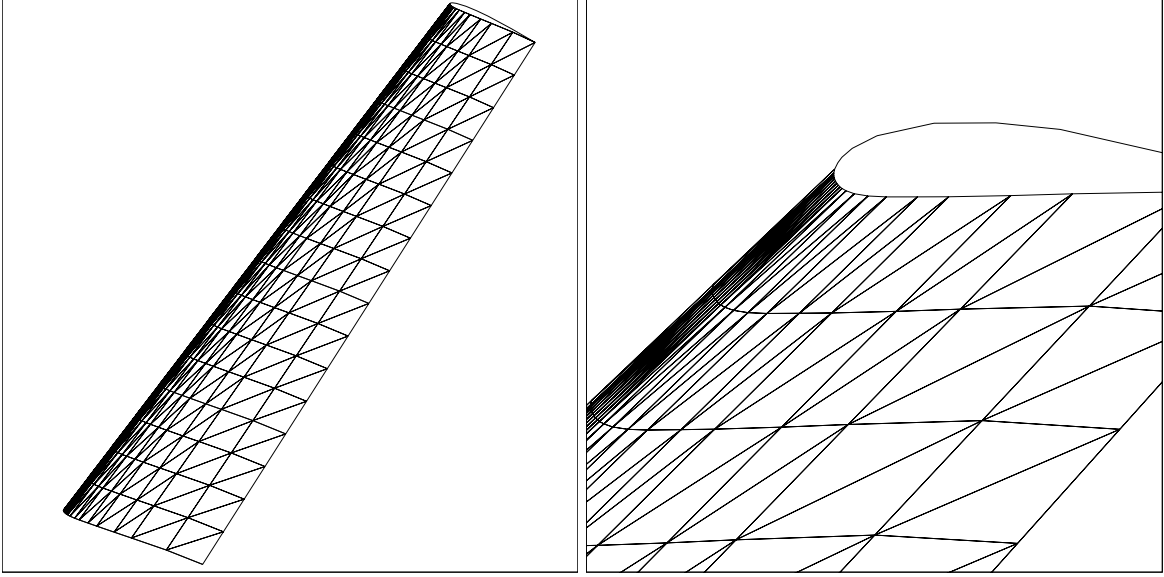


Figure 3: Triangulation based on a Trans-Finite Interpolation scheme. #points = 340, #triangles = 608, CPU time = 0.1 seconds

the underlying triangulation technique [2] is attempting to improve the minimum angle in the tessellation, the resulting triangulation is largely isotropic. While appealing for meshing entities with isotropic curvature, this approach is somewhat less than optimal for tessellating the part shown – which has a large anisotropic curvature distribution. Furthermore, the dihedral-angle based refinement rules in CAPRI’s implementation does not attempt to align inserted vertices with the direction of principle curvature. Surface bends, cylindrical trailing edges, fillets and any CAD surface with any highly anisotropic curvature require a large number of triangles to satisfy the angle metric. When combined with the isotropic nature of the MinMax triangulations of the underlying tessellation technique this misalignment can make the dihedral-angle based refinement expensive and provide results with high counts.

In contrasting the triangulations seen in Figure 1 and Figure 2 it can be noted that the tessellation of the RLV is more aesthetically pleasing. This is due to the fact that most of the triangulation was driven by a small value for the side length parameter. When producing many small triangles with approximately equal length sides (and invoking MinMax swapping) a Delauney-like triangulation is produced. In Figure 2 the transition between this length-based parameter and the curvature-based parameters can be seen in the triangles as the leading edge of the wing is approached.

Despite the high triangle counts, the CPU time required to generate the complete tessellation seen in Figure 2 of the solid (4 Faces) was only 3.8 sec¹. Nevertheless, careful analysis indicates that most of the time was spent swapping in response to site insertions triggered by the dihedral-angle criteria. Furthermore, as the number of triangles increases on a Face, the proportion of time spent swapping for surface recovery grows rapidly. While triangulation speed has been improved through the use of recursive swaps, the blind application of an essentially isotropic meshing strategy onto clearly anisotropic surface features is bound to be an expensive approach [3].

In an attempt to mitigate this problem, consider a simple Trans-Finite Interpolation (TFI) scheme applied to the quadrilateral CAD Faces on the solid. The simple wing shown in Figure 2 is composed of a number of essentially quadrilateral CAD Faces. The TFI procedure takes the (u, v) s along the bounds of the quadrilateral face and interpolates (u, v) s to interior points. These new parameter pairs can be used to evaluate to physical coordinates and therefore simply (and quickly) fill the any quadrilateral CAD Face with structured quadrilateral mesh. From this single-block quad mesh, we can quickly form triangles by simply adding diagonals to each of the quads.

This initial scheme has the following restrictions:

¹All timings in this paper are generated on a 1.8GHz Pentium 4m running LINUX.

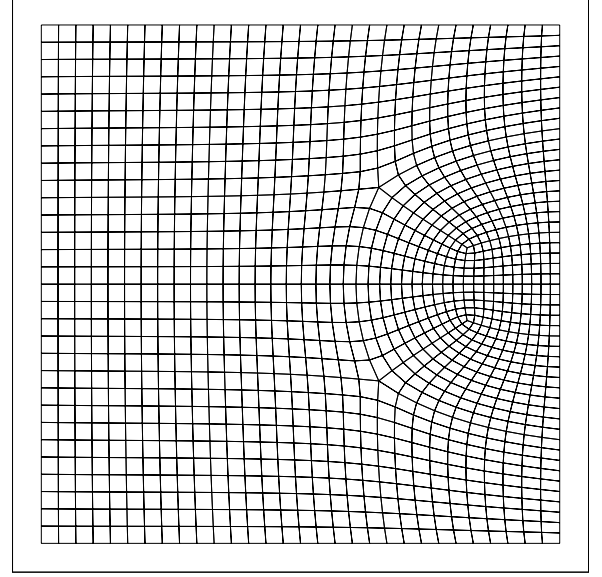
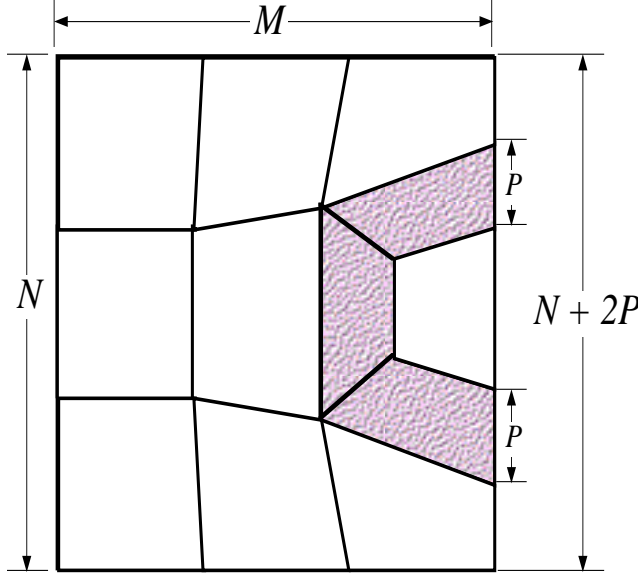


Figure 4: Generic block template and example of one set of opposite sides having the same point count.

1. Face must have only one bounding Loop (i.e. no holes)
2. The Loop must contain 4 Edges
3. Number of points found in opposing Edges must match

The Restrictions

While seemingly over-restrictive and even a step backward from the general quality triangulation approach of [3] the resulting mesh shown in Figure 3 is interesting for a number of important reasons. While the limitations listed above ensure few Faces could use such an algorithm, this simple mesh is both enlightening and encouraging. It is more regular, has significantly fewer triangles, and can be produced in a small fraction of the time required by the original approach. Moreover, the quad-based mesh still satisfies the same geometric quality metrics. The success of the algorithm exploits the fact that the four edges of the CAD Faces are aligned with or normal to the principle directions of surface curvature, and thus the underlying (u, v) s provide an efficient anisotropic ruling of these Faces.

Another less obvious advantage of this type of scheme is consistency. Any change in geometry will produce an entirely different triangulation using the standard scheme. The topology of the TFI mesh is driven only by the bounds of the quadrilateral Face. Therefore, if the point count at the Edges remain constant then the interior triangulation is consistent. This can be useful in design settings when differencing is employed

to determine parameter sensitivities. This is because point movement within a Face can be tracked.

2. UNSTRUCTURED QUADRILATERAL PATCH FILL

The easiest way to ensure a watertight triangulation of a solid is to first discretize the Edges that bound each Face. Face tessellations can then be performed using the Edge points and filling in the interior without regard to the neighboring Faces. In an attempt to capture more Faces using the TFI scheme it is obvious that the restrictions need to be relaxed.

Since we do not wish to change the manner in which the general triangulation scheme is done, it would be helpful to find a TFI-like method that does not have restriction #3. This method must also be able to produce near-normal sub-quadrilateral elements near the bounds of the Face in (u, v) so that linear features (found at the Edges) can propagate into the Face in an anisotropic manner.

Essentially, we are seeking an automatic structured meshing of any CAD face (or collection thereof), which can be reasonably morphed into a logical rectangle. The literature is rich with forays into automatic structured mesh blocking, and quadrilateral decomposition of surfaces (see for example Refs.[4]-[8]). While some of these approaches attempt a decomposition from scratch, many of the more successful methods take a “template-based” approach. When certain features are recognized in a candidate entity to be meshed, a pre-built blocking is applied. The approach outlined

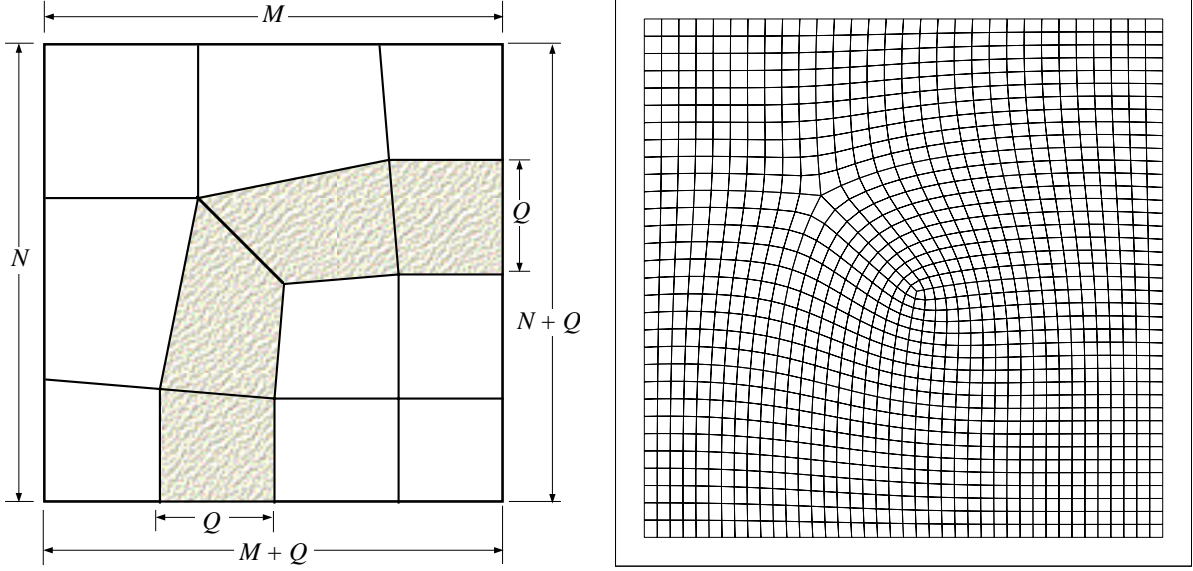


Figure 5: Generic block template and example of where opposite sides differ by the same point count (Q).

below builds upon this experience, using a generic pre-built decomposition where certain criteria are met, and using the quality technique of [3] everywhere else.

2.1 One Set of Opposite Sides Match

In this simplest case, one set of opposing quadrilateral sides match in point count – the other does not. If we assume that the largest of the mismatching sides is found at the right then the blocking that can be used to subdivide the Face is seen in Figure 4.

By examining the block template (the left side of Figure 4) one can see that the additional segments, P , in the larger side are connected back to themselves by making a loop of elements. This loop is brought back to about $1/3$ of the way in the opposite direction so that these elements do not penetrate too far to the left. Also the turning of the loop does not end up too close to the generating side so that the quads at the right side can be close to normal. The picture seen on the right-hand side of Figure 4 is a completed mesh using this block template. The size of each block is determined by either $1/3$ of the appropriate side count ($M/3$, $N/3$) or P . Note that an odd difference between the left and right sides (i.e. the long side is actually $N + 2P + 1$) is made even by reducing the vertex count on the right side by 1. This point is placed back into the final mesh by subdividing the appropriate sub-quad into 3 triangles instead of 2.

After each sub-block is populated, the result is improved by applying a Laplacian smoother.

While this example has the block oriented with the largest count on the right, this is simply a matter of convenience. The blocking template can clearly be rotated to accommodate the largest count on any of the block's perimeter Edges.

2.2 Opposite Sides Differ by a Constant

Another simple case to consider is when opposing sides differ by the same count Q . The blocking can be found in the left-hand picture of Figure 5. Again, the left side has N segments and the top has M where $N > M$. Therefore the largest side is on the right (with $N + Q$ segments) and the largest side from the 2 others can be found at the bottom ($M + Q$). Again, the orientation of the block template shown in the figure is clearly arbitrary, and the block topology applies to any rotation or reflection of the block.

For this case, it can be seen that the additional points generate elements that loop from the bottom and end up at the right quadrilateral side. The picture seen on the right-hand side of Figure 5 is a completed mesh using this block template and is constructed in a similar manner to the first case.

2.3 The General Case

The blocking for the general case can be found in the left-hand picture of Figure 6. Again, the left hand side has N segments, the top has M . In fact, this is a combination of both of the simpler cases described above.

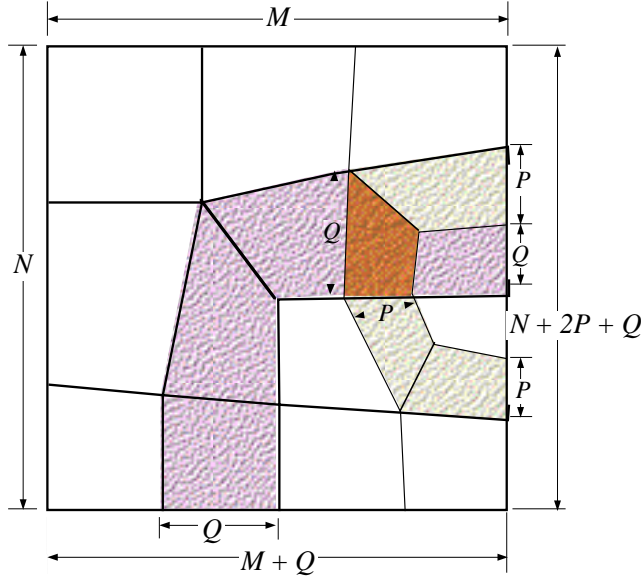


Figure 6: Generic block template and example of the General case.

17 blocks are required in order to subdivide the original quadrilateral and both simpler cases can be seen imprinted in the blocking. The circular loop is obvious on the right as depicted in Figure 4 and the set of elements coming up from the bottom can still be tracked to the right of the original quadrilateral. Here it is clear that the elements get further broken up when the circular loop intersects this group of elements.

The picture on the right-hand side of Figure 6 is a completed mesh using this block template and is constructed in a similar manner to the first case; fill each block and then apply a Laplacian smoother. One can clearly see that local orthogonality has been maintained and those places that deviate from normal to the Face sides are far from those sides. As in the earlier cases, the block template can be rotated or reflected to any orientation.

This scheme can deal with any 4 sides discretized with any number of points except for these conditions:

- A side has less than 4 points. This is because the basic method requires at least 3 blocks on a side unless the true TFI algorithm can be applied.
- The number of points on opposing sides differs greatly. It is possible to have situations where this scheme does not reduce the vertex/triangle count over the default triangulation. This occurs when there is a great disparity between opposing side counts. It has been found that when the side vertex count ratio (largest/smallest) gets greater than 3 the benefit begins to be minimized. This

heuristic is used to limit the use of the anisotropic scheme.

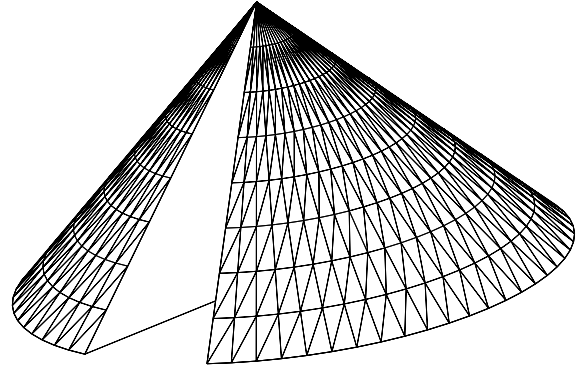


Figure 7: An example of 3 Edges on a conical surface. Note that the tip contains a degenerate (u, v) mapping.

3. LOOPS THAT DO NOT HAVE 4 EDGES

In an attempt to further remove the constraints of this TFI-like anisotropic triangulation scheme we now look at restriction #2 which constrains application of the blocking template to CAD Faces with 4 (and only 4) bounding Edges.

3.1 3 Edges

Under the limited set of circumstances that a Face with 3 Edges contains a degenerate Node; the Face

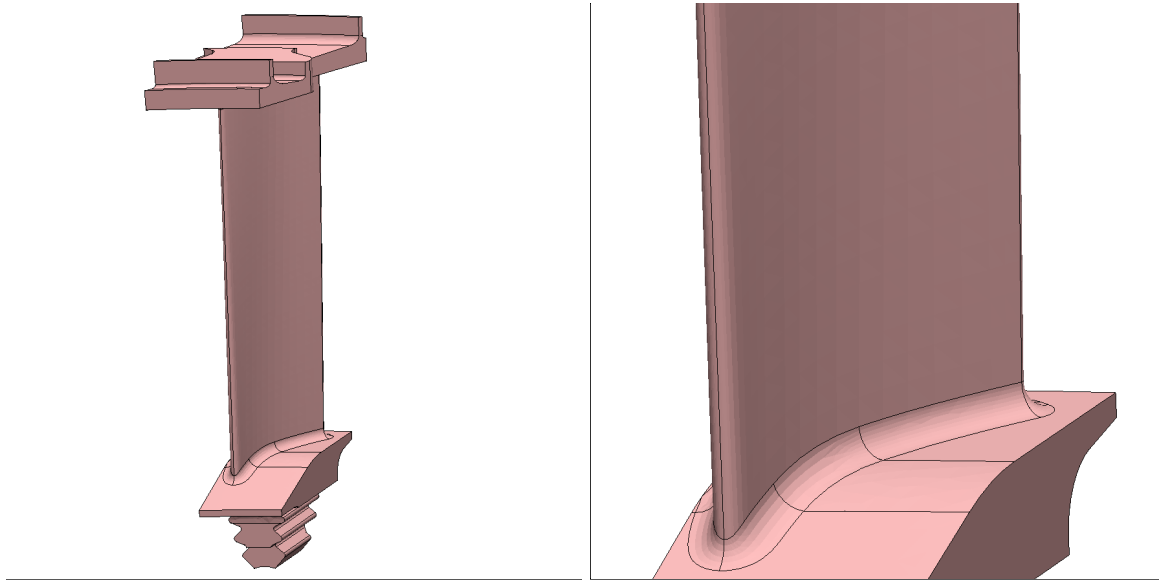


Figure 8: A turbine blade CAD part on the left. There are 94 Faces the represent the solid. On the right is a blow-up of the blade leading-edge hub junction where the fillets can be seen.

can be viewed as a quadrilateral and the technique described in the previous section can be used. Degenerate mappings can occur at the tip of a conical surface and the pole of a spherical surface. This can be identified by a discontinuity in the values of (u, v) on either side of the Node.

When this situation is found, the following technique can be used:

- Create a virtual side at the degenerate Node. A new side is created with the same number of vertices as the side that is now opposing. The (u, v) s copied from the opposite side and appropriate parameter (the one not incrementing/decrementing) is set to that found with the Node.
- Perform TFI or **One Set of Opposite Sides Match** scheme.
- Deflate the virtual side. When the sub-element quadrilaterals are broken up, any triangles that have an edge on the virtual side are not included in the final tessellation.

The results of applying this method to a split conical surface can be seen in Figure 7.

3.2 More than 4 Edges

If one can determine sets of Edges that are part of a larger continuous curve in physical space, these can

be considered a single quadrilateral side. If after analyzing all Edges there are 4 sides then the schemes described in this paper can be applied.

The technique used to examine each pair of Edges is simple and requires the following to be true:

- Is the Edge an isocline (have roughly a constant u or constant v)?
- Is the pair the same type of isocline? If so, then the pair can be considered part of a single quadrilateral side.

4. EXAMPLE – A TURBINE BLADE

The following question needs to be asked; can the methods outlined in this paper show real benefits for actual CAD parts? To answer this question we will use an example from turbomachinery. The turbine blade part contains not only the aerodynamic shape but also the hub and tip casements including the *fir tree*. The full geometry (generated in Pro/ENGINEER) can be seen in the left-hand picture of Figure 8 and a blow-up of the hub/leading-edge region can be seen on the right.

Figure 8 shows that the fillet between the hub and the aerodynamic shape as broken up into quadrilateral patches (this is seen with most all CAD systems). The set of trimming curves along the upper bounds of the fillet is a single entity as far as the aero shape

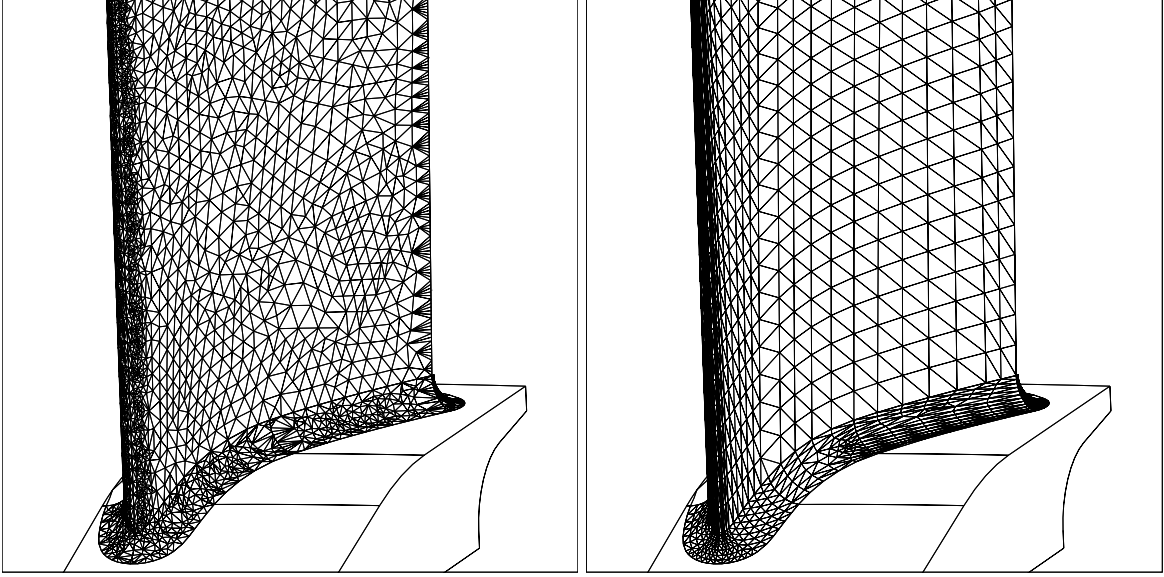


Figure 9: The surface mesh displayed on the Blade surface and some of the fillets. The picture on the left displays the results from the isotropic tessellation scheme. Seen on the right are the same Faces triangulated with the anisotropic method.

is concerned (broken up to maintain the manifold aspect of the solid). It should also be noted that the aero shape is split at the leading edge into suction and pressure surfaces each reflected in a single CAD Face.

The left-hand side of Figure 9 shows the isotropic triangulation of the blade surfaces and the fillets for the hub blown-up view. The entire tessellation of the solid contains 66308 triangles and took 60.1 CPU seconds. On the right one can see the triangulation of the same Faces (all using the anisotropic scheme). The complete solid contains 22262 triangles and took only 2.53 CPU seconds being able to treat 56 of the 94 Faces as quadrilateral patches. From the performance improvement one can assume that the Faces that consumed most of the time for the isotropic triangulation were handled by the quadrilateral scheme (the suction and pressure surfaces).

The pressure surface is properly handled even though it is bounded by more than 4 Edges (in the figure, 3 Edges mate with just the fillet Faces). An abrupt spacing change can be noted in this Face's triangulation and looks odd at first inspection. The location of this change is at the position where the fillet Faces are subdivided. Remember that with TFI, the spacing around its perimeter drives that on the interior. In this case each Edge has been discretized separately and there is much more curvature near the leading edge producing a finer set of points. The fillet Edge interior to the leading edge sees much less curvature and hence displays a coarser spacing.

The rectilinear nature of this example makes it easy to imagine that our turbine blade is unusually well suited to quadding. As a result, both the performance gain and the ratio of quadrilateral Faces to total CAD Faces are quite high in this example. In the area of performance this may be true; if the suction and pressure surfaces of the blade had cooling holes neither of the Faces could have used the anisotropic method. Nevertheless, an inexhaustive survey of several dozen CAD parts from a variety of sources with several popular geometry kernels indicates that a ratio of around 50% appears to be typical.

5. DISCUSSION

Figure 10 displays the RLV invoking the anisotropic triangulation scheme. Of the 161 CAD Faces on the 11 CAD solids, 95 of these employ the quadrilateral-blocking method discussed in this paper. The complete triangle count dropped from 750k in Figure 1 to only about 310k here, and the meshing time was reduced by approximately the same factor. The inset frames in the figure show that the quad-blocking was invoked on the vast majority of major aerodynamic surfaces. Moreover, the triangles are now well aligned along nearly all surfaces with anisotropic curvature and triangle quality and mesh smoothness are all improved over the original tessellation. This example is typical of our experience with this approach. While the approach is still limited to use on only a sub-category of CAD entities, in practice it appears to

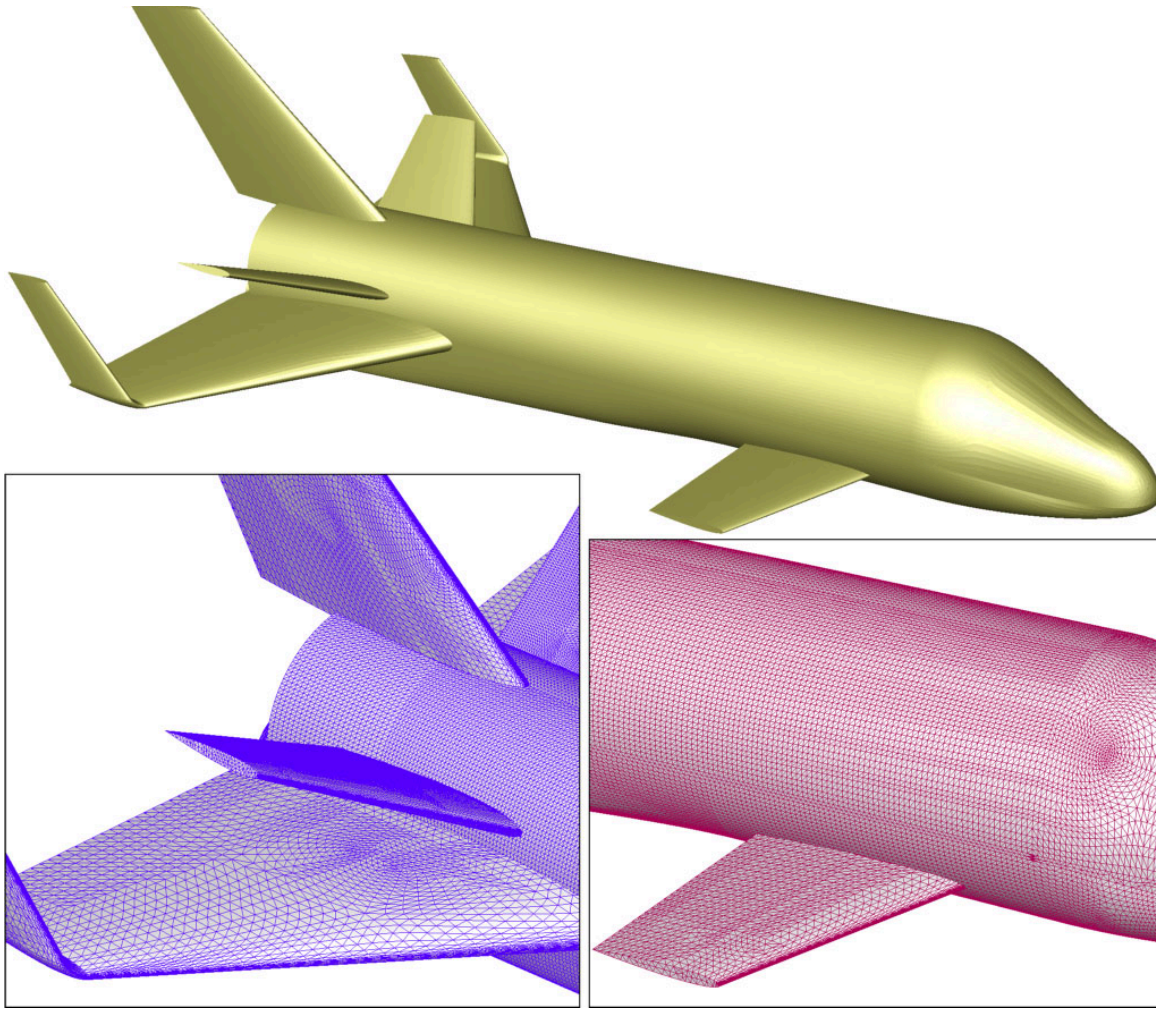


Figure 10: CAPRI's Anisotropic Triangulation for the Reusable Launch Vehicle's notional geometry.

be a very important subcategory.

The anisotropic quadrilateral patching method has been integrated into CAPRI, but unfortunately it is not fully automatic. This is due to the possible situation where the quadrilateral approach cannot be applied and the default triangulation method must be used. Due to the isotropic nature (in (u, v)) of the tessellation scheme the Edge discretization must be finer in regions of high surface curvature so the geometry can be captured. That is the default. So without *hands-on* intervention fewer CAD Faces employ the quadrilateral scheme because the constraint based on **the number of points on opposing sides differ greatly** becomes invoked.

Effort is underway to integrate the two surface meshing schemes so that it can become fully automatic. This is difficult because the Edge discretization is done before the Faces are tessellated (a requirement of the watertight attribute). The last phase of the Edge tes-

sellation is the examination of the local curvature of both of the Faces that touch the Edge. If curvature is found then the Edge tessellation continues to be enhanced. This is not only not necessary for the quadrilateral method, but can significantly reduce its quality if employed.

5.1 Resolving restriction #1

There are many quadrilateral patches that utilize the scheme presented, but those that display good quality anisotropic meshing depend on the following factors:

- The surface curvature is expressed at the Edges. This allows for the grid spacing to be set via the discretization of the quadrilateral sides. This factor is important for any Faces that contain embedded linear features such as bends, cylindrical or ruled surfaces.
- The Edges approximately follow isoclines. This

ensures that the quadrilateral patches are aligned with the curvature seen in the surface parameterization.

To remove restriction #1, one could invoke *paving* [9, 10] where general trimmed patches are broken into quadrilaterals. This would allow for the use of the templates described in this paper but would not generate anisotropic triangles based on surface features. There would be no guarantee of alignment of the patches with the underlying surface curvature.

One could image the possibility of taking the current isotropic surface tessellator and supporting anisotropic triangular meshing. One of the swapping techniques used in the triangulation scheme drives the tessellation toward isotropic (using a MinMax predicate). This is done in the underlying surface's parameter space (u, v) . Since the parameter space is artificial (i.e. not physical) any 2D mapping could be used. Therefore a transformation from a 2D space that could support an anisotropic stretching to the surfaces parameter space would be all that is required to achieve the anisotropy found in the quadrilateral patch method. There are a number of approaches that allow for anisotropic triangular meshing (including [11] and [12]). In general these schemes locally remap the space and triangulate against some isotropic predicate in the stretched space. They require a background grid or some way to get local curvature throughout the surface being meshed. This is difficult in our procedure in that we are attempting to generate the surface triangulation for the first time (and with no original reference). It is also not clear what degree of control these approaches offer in regards to mesh orthogonality.

A swapping scheme that shows promise is one that attempts to align triangle sides with the surface isoclines. The predicate looks at the sides that enclose the largest angle and swaps to minimize the deviation from (u, v) alignment. This tends to also minimize the maximum angle and would naturally be predisposed toward orthogonal meshes (in parameter space).

Finally, if performing this general anisotropic triangulation is successful, then the last phase of the Edge discretization can be removed. This then will mitigate restriction #1 giving a complete method that accurately follows CAD geometry and produces watertight tessellations with minimal counts.

ACKNOWLEDGMENTS

The authors wish to thank Peter Gage (NASA Ames Research Center) for the RLV geometry and Curran Crawford (Cambridge University) for the Turbine Blade.

The bulk of this work was performed under NASA

grant NAG2-1458.

References

- [1] R. Haimes, and G. Follen. Computational Analysis **PR**ogramming Interface. Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations. University of Greenwich, June 1998.
- [2] M. Aftosmis, M. Delanaye and R. Haimes. Automatic Generation of CFD-Ready Surface Triangulations from CAD Geometry. AIAA Paper 99-0776, January 1999.
- [3] R. Haimes, and M. Aftosmis. On Generating High Quality "Water-tight" Triangulations Directly from CAD. Proceedings of the 8th International Conference on Numerical Grid Generation in Computational Field Simulations. Honolulu HI, June 2002.
- [4] B. Kim, and S. Eberhardt. Automatic multi-block grid generation for high-lift configuration wings. NASA CP-3291, May 1995.
- [5] P. Eiseman. Multiblock grid generation with automatic zoning. NASA CP-3291, May 1995.
- [6] V.N. Vatsa, M.D. Sanetrik and E. B. Parlette. Block-structured grids for complex aerodynamic configurations: Current status. NASA CP-3291, May 1995.
- [7] F. Guibault, P. Labbe, A. Garon and R. Camarero. Automatic topographically based blocking for hybrid grid generation. AIAA Paper 98-0220, Jan. 1998.
- [8] K. Tchon, F. Guibault, J. Dompierre, P. Labbe and R. Camarero. Solution Adaptive Refinement of Multiblock Decompositions. AIAA Paper 2003-3821, Jun. 2003.
- [9] T.D. Blacker and M.B. Stephenson. Paving: A New Approach to Automated Quadrilateral Mesh Generation. IJNME, Vol. 32, 811-847, 1991.
- [10] D.R. White and P. Kinney. Redesign of the Paving Algorithm: Robustness Enhancements through Element by Element Meshing. Proceedings of the 6th International Meshing Roundtable, 1997.
- [11] F.J. Bossen, and P.S. Heckbert. A Pliant Method for Anisotropic Mesh Generation. Proceedings of the 5th International Meshing Roundtable, 1996.
- [12] K. Shimada, A. Yamada, and T. Itoh. Anisotropic Triangular Meshing of Parametric Surfaces via Close Packing of Ellipsoidal Bubbles. Proceedings of the 6th International Meshing Roundtable, 1997.